
MicroMo MVP Motion Controller

Application Note

This document describes how to configure a Paradigm operator interface terminal to allow communications with a MicroMo MVP Motion Controller. The communications protocol supports access to pertinent parameters. Please read this document carefully before attempting to configure communications with these devices.

Introduction

The EDICT-97 configuration software has been designed to allow the user to enter commands and transfer data in a manner that should be familiar to the user of a MicroMo MVP product. Items described below, set conditions that are necessary for proper communication with the MVP.

Drive Settings

The Serial "OK" response (OK) and the Checksum Calculation mode (CK) must always be set to their default state of 0. Improper operation may occur should they be set to other values.

When using RS485 (Port 3, or via a converter), the Serial Delay (SD) should be set to 10, and adjusted when necessary. This can be done via the MicroMo configuration, in which case it is recommended it be saved to EEPROM. It can also be selected in the Edict97 program, but will need to be re-sent any time the controller powers up. Incorrect settings of the Serial Delay will result in faults. This requires retries, so, less throughput.

Addressing Drives

The default Device Address, found in Comms Devices, is 0. This is broadcast mode, which transmits to all devices, but will get no response to a read request from any drive. Any other single number will expect a response to a read request, for example, AS : Report Actual Speed will wait for the drive to return the value of the speed until a timeout occurs.

The protocol in the MicroMo MVP also allows grouping drives, and the Paradigm's driver supports this function. By entering a sequence of drive addresses in the Address field of Comms Devices, separated by commas, the programmer can specify operational commands or data write operations to a group of drives. For example, one may select Device 1 with Address 1, Device 2 as Address 2, Device 3 as Address 3, and Device 4 as Address 1,3. Commands or data writes to Device 4 would then be applied by MVP units addressed as 1 and 3, and ignored by 2. Attempting to perform a Read operation (such as AS - Report Actual Speed) will result in a value of 0 being returned by the driver as the drives themselves will not generate a response.

Edict97 allows the programmer to put up to 20 characters, including the commas, in the Address field of Comms Devices. If the number of characters in the drive designations for a group exceed 20 characters, define another device with additional drives and use Edict functions such as Links, or Copy Data, to have commands transmitted to all drives.

Accessing Data

Since the MVP protocol has a portion of its descriptors as either Read, or Write, but not both, the most efficient data transfer will occur when the programmer uses the Communications Blocks rather than direct PLC References. By assigning the Read functions to blocks separately from the writes and commands, the data transfer rate will be maximized.

Identifiers specified as "Command" access requires a non-zero value in order for the transmission to be issued.

The driver described here supports the parameters in the table below.

Identifier	Description	Access
AA	Abort Action Codes	Read / Write
AB	Abort Motion Command	Command
AC	Load Profile Acceleration	Read / Write
AD	Abort Deceleration Parameter	Read / Write
AE	Auto Enable on boot	Read / Write
ANI	Analog Input Value	Read
ANM	Analog Input Mode	Read / Write
ANO	Set Current Limit	Read / Write
AS	Report Actual Speed	Read
AV	Auto Velocity on boot	Read / Write
CA	Capture A Input	Read / Write
CB	Capture B Input	Read / Write
DACA	Control DACA	Read / Write
DC	Load Profile Deceleration	Read / Write
DER	Set Derivative Loop Gain	Read / Write
DI	Disable Drive	Command
EEBOOT	Configure from EEPROM on Boot	Read / Write
EEPSAV	Save all Parameters in EEPROM	Command
EN	Enable Drive	Command
ERR	Actual Positional Error	Read
FA	Configure Following Error Action	Read / Write
FD	Set Max Dynamic Following Error	Read / Write
FDT	Set Following Error Delay	Read / Write
HA	Arm the HOME Input	Write
HF	Homing Sequence Action Code	Read / Write
HO	Define Present Position (HOME)	Write
HP	Define Home Arming Polarity	Read / Write
HS	Home Arming Status	Read
I	Set Integral Loop Gain	Read / Write
IA	Indexing Acceleration	Read / Write
IC	Index Counts	Read / Write
ID	Index Destination	Read / Write
IDC	Indexing Deceleration	Read / Write
IE	Indexing Enable	Read / Write
IM	Single/Limited/Continuous Indexing	Read / Write
IO	Index Final Destination	Read / Write
IS	Indexing Velocity	Read / Write

Identifier	Description	Access
ITD	Index Destination Delay	Read / Write
ITR	Index Serial Trigger	Command
ITZ	Index Zero Delay	Read / Write
J	Set Local Velocity Range Multiplier	Read / Write
JH	Set Joystick Hysteresis Window	Read / Write
JM	Select Joystick Mid Point	Read / Write
JW	Select Joystick Center Window	Read / Write
K	Select Continuous Integration	Read / Write
LA	Load Absolute Target Position	Read / Write
LL	Set Position Range Limits	Read / Write
LP	Define Limit and E-Stop Polarity	Read / Write
LR	Load Relative Target Position	Read / Write
LS	Limit Sequence Enable	Read / Write
M	Initiate Motion	Command
ME	Macro Execute	Read / Write
MS	Macro Status	Read
N	Define the "In Position" Range	Read / Write
O	Local/Remote Mode Flag	Read / Write
POR	Proportional Loop Gain	Read / Write
POS	Present Position	Read
POSCA	Capture Position A	Read
POSCB	Capture Position B	Read
PWM	Set Minimum PWM Duty Cycle	Read / Write
RD	Reverse Operational Direction	Read / Write
RE	Reverse Encoder Phasing	Read / Write
RN	Reset Node	Command
SA	Select Range Limit Action	Read / Write
SD	Serial Response Delay (Active on RS485 only)	Write
SP	Load Max. Commanded Velocity	Read / Write
SR	Set MVP Loop Sample Period	Read / Write
ST	Present Node Status	Read
T	Set Percentage Trajectory Parameter	Read / Write
V	Constant Velocity Command	Read / Write
VLIM	Control Output Voltage Limit	Read / Write
W	Define the Open Loop PWM Duty	Read / Write
X	Synchronize Nodes	Command

Knowledge of Unit Operation Is Assumed

In all cases, the simple principle of 'pass-through' is maintained: there is no attempt to validate a value in terms of the end use of the unit: both familiarity with the Drive functions and knowledge of system operation are assumed.

Communications

RS232 Connection Table

Paradigm Port 2 (RS232)	MicroMo MVP
1	2
2	3
5	5

Additionally, connect a jumper between terminals 3 and 4 of the Paradigm.

RS485 Connection Table

Paradigm Port 3 (RS485)	MicroMo MVP
6 connected to 8	3
7 connected to 9	2
10	5

Additionally, connect a 1K resistor between 9 and 10 of the Paradigm.